

Implementing quantum gates by optimal control with doubly exponential convergence

Pierre de Fouquieres

*Centre for Quantum Information and Foundations,
Department of Applied Maths and Theoretical Physics, University of Cambridge,
Wilberforce Road, Cambridge, CB3 0WA, United Kingdom*

(Dated: February 14, 2012)

We introduce a novel algorithm for the task of coherently controlling a quantum mechanical system to implement any chosen unitary dynamics. It performs faster than existing state of the art methods by one to three orders of magnitude (depending on which one we compare to), particularly for quantum information processing purposes. This substantially enhances the ability to both study the control capabilities of physical systems within their coherence times, and constrain solutions for control tasks to lie within experimentally feasible regions. Natural extensions of the algorithm are also discussed.

PACS numbers: 03.67.Ac, 02.30.Yy, 03.67.Lx, 03.65.Yz

The coherent control of quantum mechanical systems [1, 2] has been successfully applied to a growing number of tasks in recent years [3, 4]. The early approaches such as two pathway quantum interference, pump-dump schemes, or stimulated Raman adiabatic passage are intrinsically understandable in terms of interference due to the coordinated activation of resonant transitions between few energy levels [5]. In order to extend such strategies and tackle more challenging problems, the field has moved towards employing pulse shapers and optimization algorithms [6].

The field has also broadened its scope, from problems of state preparation, or more generally maximizing the value of an observable over an ensemble [7], to, in particular, implementing unitary maps [8]. This latter bridges the gap between physical dynamics and the gate formalism of quantum information processing, for which high accuracy solutions are sought, ultimately aiming to reach an error correction threshold around 10^{-3} to 10^{-4} per gate or below [9]. In applying the same methods to both state, and map or gate problems, the additional structure inherent to gate problems has however been neglected – the aim of this letter is to describe how this structure can be exploited to better understand and substantially ease the solving of gate problems.

Formally, a closed N -level system undergoes controlled unitary dynamics given by $U_{\mathbf{f}}$ satisfying

$$\frac{\partial}{\partial t} U_{\mathbf{f}}(t) = -iH[\mathbf{f}(t)]U_{\mathbf{f}}(t), \quad U_{\mathbf{f}}(0) = I$$

with I the identity matrix, and a time dependent Hamiltonian

$$H[\mathbf{f}(t)] = H_0 + \sum_{r=1}^R \mathbf{f}_r(t) H_r \quad (1)$$

where \mathbf{f} is a set of R controls pulses, altering the system potential within a semi-classical model under the bilinear approximation. The abstract control problem for a target unitary gate V consists in finding a set of real

valued functions \mathbf{f} and an evolution time T such that the dynamics satisfies $U_{\mathbf{f}}(T) = V$. Since this entails transferring a full basis of states to another (with relative phases), the intuition applicable to state problems is no longer available, and in fact schemes to solve it explicitly are limited to two level systems, or special cases with few levels. For practical purposes, we would only require the actual $U_{\mathbf{f}}(T)$ and target V to match up to some prescribed error level ε , with respect to a notion of distance d . Optimisation, whereby a sequence of pulses $\mathbf{f}^{(n)}$ is generated iteratively with the requisite distance $d(U_{\mathbf{f}^{(n)}}(T), V)$ decreasing at each step, has emerged as the strategy of choice for achieving this. The definitions of distance $d(U, V)$ to measure the error have generally been based on the Hilbert-Schmidt norm, using either $\|U - V\|$ or, quotienting out the unphysical global phase of the dynamics and normalising,

$$d(U, V) = \frac{1}{2\sqrt{N}} \min_{\varphi} \|U - e^{i\varphi} V\|$$

which we will be using herein. In contrast to state control problems where intuitive understanding often plays a role in choosing the initial trial pulses $\mathbf{f}^{(0)}$, the serious limitations of intuitive insight for gate problems lead to $\mathbf{f}^{(0)}$ typically being chosen arbitrarily, eg. at random. Moreover, to get any measure of gate error based on experimental measurement would require exhaustive and arduous process tomography, so that there has been an overwhelming preference towards working with numerical simulation of a model for the system.

Running a numerical optimisation algorithm requires that the control pulses be discretised, and in order to incorporate experimental constraints we can choose a basis for discretisation corresponding to the capabilities of our pulse shaping equipment. Thus we let

$$\mathbf{f}_r(t) = \sum_{k=1}^K \alpha_{rk} b_k(t)$$

and then optimise over the set of RK coefficients α_{rk} ; ide-

ally the basis elements b_k would be precisely calibrated to the equipment, but for definiteness we will consider representatives of two important cases. A common choice in the literature, and the main one we will use, is that of piecewise constant functions, as can be produced by an arbitrary waveform generator [10]. In the case of frequency domain pulse shaping, we are dealing with functions which, up to Gaussian tails, are both time and spectral bandwidth limited. To capture this property, under suitable scaling we can let b_{k+1} be the Hermite function [28] of index k , and restrict to the first K of these. Such a choice has on the other hand not been used in the quantum control setting to our knowledge, although the benefits of Hermite functions have certainly been exploited in other applications, eg. [11] – while the smoother alternatives to piecewise constant functions used, such as truncated interpolating polynomials [12], have much fatter tails in frequency domain. In addition to the basis constraint on the pulses \mathbf{f} , there must clearly be some bound B on the pulse fluences, equivalently their magnitude in the integrated power norm.

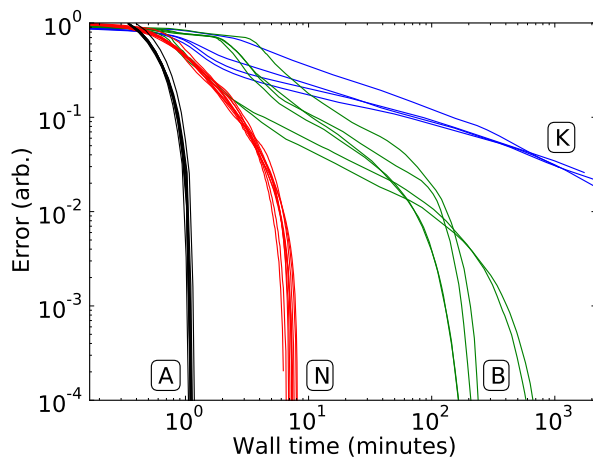


FIG. 1: Performance comparison between several runs of the Newton-Raphson (red, labeled ‘N’), BFGS GRAPE (green, labeled ‘B’) and Krotov (blue, labeled ‘K’) algorithms with small initial pulses $\mathbf{f}^{(0)}$. Also shown (in black, labeled ‘A’) are Newton-Raphson runs initialised at the norm with least ill-conditioning; the cost of finding this norm, on average 21 seconds, is included.

For generic intrinsic and control Hamiltonians H_0, \dots, H_R of (1), as well as many specific cases of physical interest, full controllability is known to hold, meaning that any target gate can be achieved given sufficient evolution time T and freedom in shaping the control pulses [13]. While this is a strong result, it does not identify which gates are achievable for particular evolution times and constraints on the controls, in any given experimental context. Indeed, such specific results are currently lacking, and one must resort to numerical investigations

in order to gain better understanding into the capabilities of each physical system [14]. This motivates the need for an optimisation algorithm to solve the gate problem in runtimes on a scale rendering the process interactive or faster. It is the purpose of this letter to introduce a Newton-Raphson root finding approach for this problem, which performs substantially faster than existing methods (see Fig. 1), thereby achieving this goal on non-trivial examples. Such an approach also sheds light on the strong influence of the pulse initialisation $\mathbf{f}^{(0)}$, leading to a prescription for how to choose it.

For consistency, all numerical examples herein are for the canonical problem of implementing a quantum Fourier transform on a five qubit Ising coupled spin chain in a magnetic field gradient using two controls. But these are qualitatively representative of the results for different evolution times, target gates, and modes of control – a different problem scenario illustrates such similarity in the supplement. In terms of Pauli matrices σ , the Hamiltonian in question is

$$H[\mathbf{f}(t)] = \sum_{n=1}^4 \sigma_z^{(n)} \sigma_z^{(n+1)} - \sum_{n=1}^5 (n+2) \sigma_z^{(n)} + \mathbf{f}_1(t) \sum_{n=1}^5 \sigma_x^{(n)} + \mathbf{f}_2(t) \sum_{n=1}^5 \sigma_y^{(n)} \quad (2)$$

while we fix an evolution time T of 125 and use $K = 1000$ basis functions, with piecewise constant controls unless stated otherwise.

Over the last fifteen years, most techniques successfully applied to model-based quantum control problems have either come from mainstream gradient-driven optimisation theory, eg. conjugate gradient and BFGS [15] based GRAPE [16] algorithms, or can be understood in this context, as with the Krotov method [17]. These state of the art techniques have led to advances such as towards implementing logic gates fault-tolerantly [18] or with minimal errors given the decoherence time [19]. They owe their performance to the use of gradient information, but a key realisation is that the full Jacobian matrix \tilde{J} of $U_{\mathbf{f}}(T)$ for the gate problem can be computed as efficiently as its single row constituting the gradient vector. Indeed the usual gradient computation [20], for $\|U_{\mathbf{f}}(T) - V\|^2$ say, effectively proceeds through \tilde{J} by inner producting each row of \tilde{J} with V , so that using the gradient alone means discarding a lot of valuable information.

Looking at the singular value decomposition of the Jacobian matrix leads to a clean geometric picture, whereby changes to the pulses below a certain norm r (beyond which higher order terms cease being negligible) induce changes in the implemented gate within a prescribed ellipsoid. The basic Newton-Raphson iteration [21] then consists in using this Jacobian model, with a heuristic choice of r , to compute new pulses bringing the imple-

mented $U(T)$ closer to the target V , which reduces to a linear algebraic task. In order to have the modelling ellipsoid strictly track the unitary group, one can map its elements down via the matrix exponential, or conversely, group elements up via the matrix logarithm, as we describe later. The volume of this ellipsoid determines the ability of all algorithms mentioned herein to shift $U_f(T)$ in general directions, while for the specific target V a more relevant quantity correlated to this volume is the distance to exact solution controls upon ignoring higher order terms, which we shall refer to as the level of ill-conditioning.

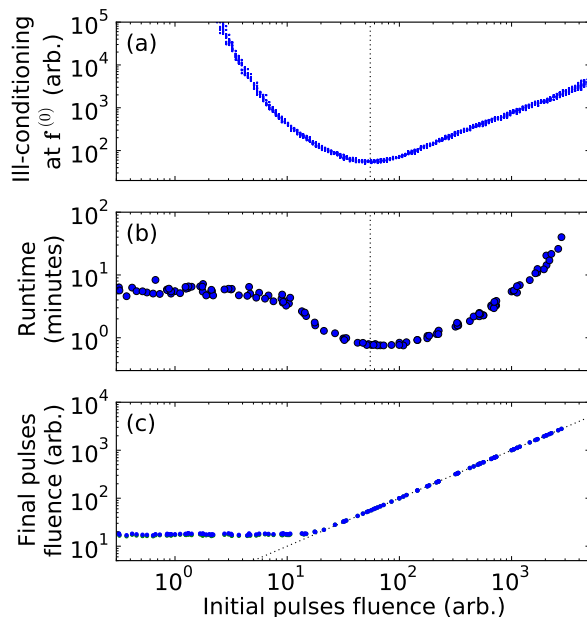


FIG. 2: For Newton-Raphson runs with initial pulses $\mathbf{f}^{(0)}$ of different norms, (b) the wall time needed to reach an error ε of 10^{-4} and (c) the norm of the corresponding solution pulses, with a dashed ‘initial equals final’ line. In addition, (a) the ill-conditioning of the Jacobian at several randomly sampled pulses of each norm.

The situation for our test problem depicted in Fig. 2 is representative of the general structure, as described in detail in the supplemental material, which can be expected of all problems. We see that both the ill-conditioning and the Newton-Raphson runtime strongly depend on the integrated power of the pulses (specifically of $\mathbf{f}^{(0)}$ in the latter case), but are well concentrated beyond this along a single curve, with the minimum of these two curves coinciding. When solution pulses are required to have a fluence below B , we can cheaply find the minimum of the ill-conditioning curve restricted to the interval $[0, 4B/5]$ say, and use an arbitrary $\mathbf{f}^{(0)}$ with this norm to initialise the algorithm. It is clear from Fig. 2(c) that this choice will yield a solution satisfying the fluence constraint unless no choice can, while the correspondance between ill-

conditioning and runtime curves makes it the most efficient choice. The benefit of this prescription over less deliberate ones is evident from comparing the ‘A’ and ‘N’ series of runs in Fig. 1.

The original equation $U_f(T) = V$ should be thought of as over-determined when the dimension N^2 of the unitary group is greater than that of the control space RK , since then it is only solvable to arbitrarily high accuracy for an exceptional set of targets V . This makes it an unfavourable case in the context of low error control, because it is implausible for a target gate V of interest to be special in this sense. On the other hand, whenever solvability is not so limited then almost every achievable target V admits an $RK - N^2$ dimensional set of control pulses implementing it. Although the number of iterations for algorithms to reach a given error tolerance ε is, as expected, reduced as this dimension of degeneracy increases, there is a counter-intuitive downside to under-determined problems.

In general when converging to an exact solution, the error of Krotov iterates decays exponentially, ie. eventually as γ^n for some $\gamma < 1$, while with conjugate gradient or BFGS, the error decay is faster than exponential, and Newton algorithms have error decaying doubly exponentially [22], as $O(\beta^{2^n})$ for some β . But in the under-determined context, one can only count on exponential convergence from all of these algorithms except for Newton-Raphson root finding which retains its double exponential convergence. Indeed, the directions of degeneracy about a solution form a null space to the Hessian there, rendering inapplicable the analysis [23] on which faster than exponential convergence results for BFGS are based [24, 25]. The stark difference between these rates is illustrated in Fig. 3, where all Newton-Raphson runs surpass 10^{-4} in a single iteration once they reach 10^{-2} error.

In order to make best use of Newton-Raphson root finding, we must re-formulate our problem over a linear space, and the most natural choice here is to seek for the functional

$$\mathcal{L}(\mathbf{f}) = \log(V^\dagger U_f(T))$$

to equal zero within the space \mathfrak{u} of anti-Hermitian matrices. The resulting algorithm then has the elegant property of reducing the geodesic distance between the actual $U_f(T)$ and target gate V on each iteration. It can usefully be made more general by restricting attention to a subspace of \mathfrak{u} , specified by an orthogonal projection P , ie. seeking a zero of $P\mathcal{L}(\mathbf{f})$. In particular, restricting to the space \mathfrak{su} of traceless anti-Hermitian matrices makes the root finding insensitive to the unphysical global phase.

A less obvious application is to implement a gate on a system interacting coherently with an environment [26], which contrary to Markovian interaction with a bath is reversible so need not fundamentally limit the achievable error. The full Hilbert space then splits as $S \otimes E$, and

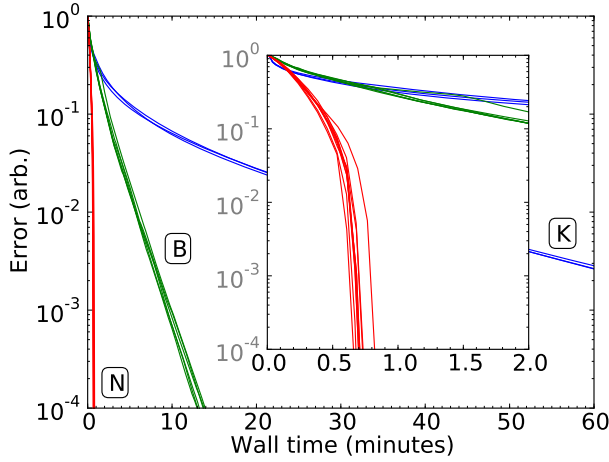


FIG. 3: Illustration of the different convergence behaviours of Newton-Raphson (red, labeled ‘N’), BFGS GRAPE (green, labeled ‘B’) and Krotov (blue, labeled ‘K’) algorithms with initial norms having minimal ill-conditioning. Although several runs were carried out from very different initial pulses, the performance profile of each algorithm remains the same.

for a given gate on the system W our aim would be to implement any gate of the form $W \otimes A$ for an ancillary evolution A – this corresponds to letting $V = W \otimes I$ in \mathfrak{L} and projecting it out of the space $I \otimes \mathfrak{u}$ with P . Although the Newton-Raphson algorithm can certainly be applied to Lindblad dynamics, choosing a short evolution time T to limit dissipation and finding a control for the system without bath should still be a first step, since computing the evolution super-operator is much more expensive. In both extensions, having $RK \geq N^2$ becomes far from sufficient to justify concluding the problem is exactly solvable.

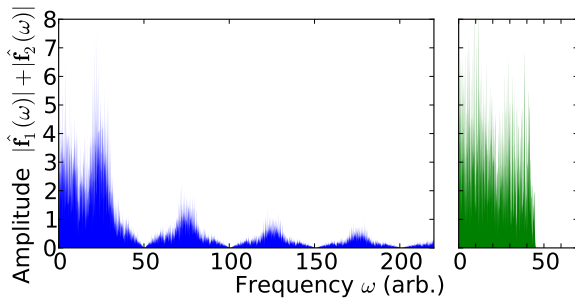


FIG. 4: Spectrum of typical minimal norm solutions with error below 10^{-4} , in the piecewise constant (left) and Hermite function (right) bases. These are symmetric about $\omega = 0$, but the Hermite functions could just as easily be made bandlimited about any chosen carrier frequency.

Up until now, all numerical examples have been in the piecewise constant basis, but our algorithm applies to general bases, and of particular interest is the Hermite

basis to find spectrally narrow solution pulses. Note that the same bandwidth as seen in the right panel of Fig. 4 could be obtained by a suitable basis of low frequency Fourier components, but implementing such a pulse in a finite time duration would lead to distortion, thereby deteriorating the achieved error. Interestingly, using the same number of basis functions K , the curve from Fig. 2(c) and the most efficient initial fluence remain unchanged across both bases – the average number of iterations to reach the same error tolerance ε is also similar (10.1 vs. 12.5) for this initial fluence. The piecewise constant basis is however special in how operations are cheaper with it than in general bases, eg. for our test problem computing the propagator $U_f(T)$ takes 0.9 seconds, as opposed to around 50 seconds in the Hermite basis. This makes it all the more important to choose the initial $\mathbf{f}^{(0)}$ in a general basis carefully, and to this end information from the more tractable piecewise constant case seems to suffice.

We have seen how viewing unitary map control problems from the root finding perspective motivates an algorithm offering vast performance improvements over existing methods, and reveals particularly clean structure within the space of controls. This formulation can moreover naturally be made in the full generality of pulses represented in arbitrary bases and accounting for an environment to the system. For state preparation problems, considering the Newton-Raphson algorithm analogue [27] in that case promises to lead to further fruitful developments.

This work was funded by EPSRC, via CASE/CNA/07/47, and Hitachi. The author wishes to thank Sophie Schirmer and Peter Pemberton-Ross for valuable exchanges.

-
- [1] H. M. Wiseman and G. J. Milburn. *Quantum measurement and control*. (Cambridge University Press, 2009)
 - [2] M. Shapiro and P. Brumer. *Principles of the quantum control of molecular processes*. (Wiley-Interscience, 2003)
 - [3] J. P. Dowling and G. J. Milburn. *Phil. Trans. R. Soc. London A*, **361**, 1655 (2003)
 - [4] H. Mabuchi and N. Khaneja. *Int. J. Robust & Nonlin. Cont.*, **15**, 647 (2005)
 - [5] C. Brif, R. Chakrabarti, and H. Rabitz. *New J. Phys.*, **12**, 075008 (2010)
 - [6] K. Singer, U. Poschinger, M. Murphy, P. Ivanov, F. Ziesel, T. Calarco, and F. Schmidt-Kaler. *Rev. Mod. Phys.*, **82**, 2609 (2010)
 - [7] D. J. Tannor and S. A. Rice. *J. Chem. Phys.*, **83**, 5013 (1985)
 - [8] T. Schulte-Herbrüggen, A. Spörl, N. Khaneja, and S. J. Glaser. *Phys. Rev. A*, **72**, 042331 (2005)
 - [9] A. M. Steane. *Phys. Rev. A*, **68**, 042322 (2003)
 - [10] F. Motzoi, J. M. Gambetta, S. T. Merkel, and F. K. Wilhelm. *arXiv:1102.0584*, (2011)
 - [11] P. K. Higuchi and S. Karp. *U.S. Patent* 3384715 (1968)

- [12] J.-S. Li, J. Ruths, T.-Y. Yu, H. Arthanari, and G. Wigner. *P. Nat. Acad. Sci.*, **108**, 1879 (2011)
- [13] V. Jurdjevic and H. J. Sussmann. *J. Diff. Eq.*, **12**, 313 (1972)
- [14] T. Caneva, M. Murphy, T. Calarco, R. Fazio, S. Montanero, V. Giovannetti, and G. E. Santoro. *Phys. Rev. Lett.*, **103**, 240501 (2009)
- [15] P. de Fouquieres, S. G. Schirmer, S. J. Glaser, and I. Kuprov. *J. Mag. Res.*, **212**, 412 (2011)
- [16] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S.J. Glaser. *J. Mag. Res.*, **172**, 296 (2005)
- [17] Y. Maday and G. Turinici. *J. Chem. Phys.*, **118**, 8191 (2003)
- [18] R. Nigmatullin and S G Schirmer. *New J. Phys.*, **11**, 105032 (2009)
- [19] A. Spörl, T. Schulte-Herbrüggen, S. J. Glaser, V. Bergholm, M. J. Storz, J. Ferber, and F. K. Wilhelm. *Phys. Rev. A*, **75**, 012302 (2007)
- [20] I. Kuprov and C. T. Rodgers. *J. Chem. Phys.*, **131**, 234108 (2009)
- [21] C. T. Kelley. *Solving nonlinear equations with Newton's method*. (SIAM, 2003)
- [22] A. Ben-Israel. *J. Math. Anal. & Appl.*, **15**, 243 (1966)
- [23] J. E Dennis and J. J Moré. *Math. of Comp.*, **28**, 549 (1974)
- [24] M.J.D. Powell. *SIAM-AMS P. Nonlin. Prog.*, **9**, 53 (1976)
- [25] D. H Li and M. Fukushima. *SIAM J. Optim.*, **11**, 1054 (2001)
- [26] P. Rebentrost, I. Serban, T. Schulte-Herbrüggen, and F. K. Wilhelm. *Phys. Rev. Lett.*, **102**, 090401 (2009)
- [27] J. Martínez. *J. Comp. & Appl. Math.*, **34**, 171 (1991)
- [28] Also known as the eigenfunctions of the harmonic oscillator

SUPPLEMENTAL MATERIAL

We will now describe the algorithm as implemented for producing numerical results in this work, aiming to include sufficient technical detail that it may be fully replicated based on this account. This specification focuses on the resulting performance more than ease of implementation, and most of its prescriptions can be legitimately simplified as needed. A source code release, to be made available as a Python SciKit[17] under the code-name CUMIN, is also in preparation.

Numerical optimisation

A common thread in derivative-based optimisation [1], aiming to minimise an objective function $\mathfrak{E} : \mathbb{R}^M \rightarrow \mathbb{R}$, is the repeated update of a variable \mathbf{a} storing what can be thought of as the current base point for the domain \mathbb{R}^M . We will write $\mathbf{a}^{(0)}, \mathbf{a}^{(1)}, \dots$ for the sequence of values taken by \mathbf{a} , but omit the superscript when referring to its value at some present state of the algorithm. On each iteration, say the n^{th} , a model for \mathfrak{E} valid locally about the outgoing base point $\mathbf{a}^{(n-1)}$ for this iteration is constructed based on the value of certain derivatives

of function \mathfrak{E} at the point $\mathbf{a}^{(n-1)}$. The idea is then to take for the next base point $\mathbf{a}^{(n)}$ the minimum \mathbf{v} of this model for \mathfrak{E} , possibly restricted to some neighbourhood of $\mathbf{a}^{(n-1)}$. However, since the model is only valid locally, we cannot guarantee $\mathfrak{E}(\mathbf{v}) < \mathfrak{E}(\mathbf{a}^{(n-1)})$ ie. that using \mathbf{v} as the next base point will lead to a reduction of the current objective value $\mathfrak{E}(\mathbf{a})$. Two families of approaches exist for resolving this problem: line search methods look for a suitable point along the line starting at $\mathbf{a}^{(n-1)}$ in the direction of \mathbf{v} , while trust region methods consider rescaling the neighbourhood of $\mathbf{a}^{(n-1)}$ to which they constrain minimisation of the model for \mathfrak{E} .

The main performance characteristic which can be reliably affected in choosing between derivative-based optimisation algorithms is the run-time required to achieve convergence to a given accuracy. When methods have different asymptotic rates of convergence, one of them is determined in advance to eventually be closer to the limit than all others. Prior to this regime, there is a trade-off in the accuracy of models used on each iteration, where more derivative information can be acquired at higher computational cost to construct better models, which should in contrast enable greater progress per iteration. In this respect, the Hessian matrix of all $M(M+1)/2$ second derivatives is prohibitively expensive to compute except for special \mathfrak{E} , so that widely effective algorithms only require the gradient vector of M first derivatives.

For example quasi-Newton optimisation methods, of which BFGS is an instance, use the model $\mathfrak{E}(\mathbf{a} + \mathbf{p}) \simeq \mathfrak{E}(\mathbf{a}) + \mathbf{g}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{B} \mathbf{p}$ where \mathbf{g} is the gradient vector of \mathfrak{E} at \mathbf{a} , and \mathbf{B} is constructed iteratively to approximate the Hessian matrix of \mathfrak{E} at \mathbf{a} . At least for BFGS, the model minimum \mathbf{v} , namely $-\mathbf{B}^{-1} \mathbf{g}$ since \mathbf{B} is always positive definite, is used within a line search routine at each iteration. In this context, steepest descent would correspond to replacing \mathbf{B} by the identity matrix \mathbf{I} , but we do not consider this method since it cannot be recommended in general, and performs particularly slowly on gate control problems [2]. For comparison, convergence of BFGS iterates to a local minimum \mathbf{a}^* where the Hessian of \mathfrak{E} is positive definite is understood to happen at a rate, measured either by $\|\mathbf{a}^{(n)} - \mathbf{a}^*\|$ or $\mathfrak{E}(\mathbf{a}^{(n)}) - \mathfrak{E}(\mathbf{a}^*)$, between $O(\xi^{n \log n})$ [3] and $\Omega(\xi^{n^2})$ [4] for some problem specific ξ . While convergence for conjugate gradient [5] and limited memory BFGS [6], without being restarted every M iterations only goes as ξ^n asymptotically, which is also the rate for steepest descent albeit with ξ deteriorating significantly [7] on poorly conditioned problems.

Newton-Raphson method

This description can be adapted to cover Newton-Raphson root finding applied to the vector valued function $\mathfrak{L} : \mathbb{R}^M \rightarrow \mathbb{R}^m$ by introducing an objective function $\mathfrak{E}(\mathbf{x}) = \|\mathfrak{L}(\mathbf{x})\|^2$ and specifying that the model, hence

the derivatives used to construct it, should be of \mathcal{L} rather than \mathcal{E} . The model in this case reads

$$\mathcal{L}(\mathbf{a} + \mathbf{p}) \simeq \mathcal{L}(\mathbf{a}) + J\mathbf{p}$$

where J is the Jacobian matrix of \mathcal{L} at the point \mathbf{a} , with entry i, j equal to the j^{th} partial derivative $\frac{\partial}{\partial x_j}$ of the i^{th} component of $\mathcal{L}(\mathbf{x})$ – in the notation of differential geometry, we can succinctly write J as $d\mathcal{L}|_{\mathbf{a}}$.

If the model were exact, assuming the rank of J equals m , a global minimum of \mathcal{E} could be found where \mathcal{L} is $\mathbf{0}$, namely at $\mathbf{a} + \mathbf{p}$ with \mathbf{p} being any solution to $J\mathbf{p} = -\mathcal{L}(\mathbf{a})$. Indeed the classical Newton-Raphson algorithm, applicable for dimensions $M = m$, uses the update rule $\mathbf{a}^{(n+1)} = \mathbf{a}^{(n)} + \mathbf{p}$ for which $\mathcal{E}(\mathbf{a}^{(n)})$ goes to zero doubly exponentially in n , specifically as $O(\beta^{2^n})$ for some β (known as quadratic convergence), for suitable initial conditions $\mathbf{a}^{(0)}$. Generalising this to the under-determined regime $M > m$, the value of \mathbf{p} is no longer explicitly determined as $-J^{-1}\mathcal{L}(\mathbf{a})$, and a suitable choice for \mathbf{p} which preserves the quadratic convergence property is the minimum norm solution of $J\mathbf{p} = -\mathcal{L}(\mathbf{a})$ [8], computable as $-J^T(JJ^T)^{-1}\mathcal{L}(\mathbf{a})$. In either case, a suitable initial $\mathbf{a}^{(0)}$ would be any point sufficiently close to some solution \mathbf{a}^* of $\mathcal{L}(\mathbf{a}^*) = \mathbf{0}$ where the Jacobian $d\mathcal{L}|_{\mathbf{a}^*}$ has full rank m , in such a way that $d\mathcal{L}|_{\mathbf{a}^{(n)}}$ is always full rank so that all iterates $\mathbf{a}^{(n)}$ are well-defined.

Finding such an initial point is itself difficult, and for general points \mathbf{a} the neighbourhood in which the model at \mathbf{a} is accurate contains neither a root \mathbf{a}^* of \mathcal{L} , nor a root $\mathbf{a} + \mathbf{p}$ of the model. The rationale behind the update rule $\mathbf{a}^{(n+1)} = \mathbf{a}^{(n)} + \mathbf{p}$, that $\mathbf{a}^{(n+1)}$ should be close to a root of \mathcal{L} since $\mathcal{L}(\mathbf{a}^{(n)} + \mathbf{p}) \approx \mathbf{0}$, therefore no longer applies for general $\mathbf{a}^{(n)}$ and we are compelled to invoke a line search or trust region method to find a usable $\mathbf{a}^{(n+1)}$. But before we delve further into this, let us deal more specifically with the function \mathcal{L} which we use for the unitary map control problem. Note in passing that for a general objective \mathcal{E} which is strictly convex at some minimiser \mathbf{a}^* , applying Newton-Raphson to its gradient vector $\mathbf{g}(\mathbf{x}) = d\mathcal{E}|_{\mathbf{x}}$ will seek a critical point of \mathcal{E} so must converge to \mathbf{a}^* when starting sufficiently close to it. The direction of line searches would then be $-J^{-1}\mathbf{g}$ where J is the Hessian matrix of \mathcal{E} , justifying the approximating $-B^{-1}\mathbf{g}$ used in BFGS.

Unitary map problem

Consider the dynamical Lie algebra \mathfrak{l} of our control system, ie. the linear space spanned by all iterated commutator expressions starting with the matrices iH_0, iH_1, \dots, iH_R . By the Frobenius theorem, the propagators $U_{\mathbf{f}}(t)$ must remain within the associated group $e^{\mathfrak{l}}$, consisting of matrix exponentials of matrices in \mathfrak{l} , for all time and over all possible control vectors (f_1, \dots, f_R) .

Conversely, the controllability theory of bi-linear systems shows that for compact $e^{\mathfrak{l}}$ there is a critical time T_c depending only on the system such that every point of $e^{\mathfrak{l}}$ is accessible in some time $T < T_c$ using some control vector \mathbf{f} . This result also holds if the class of admissible controls is restricted to both smooth or piecewise constant functions. At least when in addition \mathfrak{l} is semi-simple [9], the set of unitary matrices $U_{\mathbf{f}}(T)$ which are accessible at a fixed time T using some control vector \mathbf{f} has non-empty interior within $e^{\mathfrak{l}}$ [10], for each $T > 0$, and in fact equals $e^{\mathfrak{l}}$ for T sufficiently large. The conditions of $e^{\mathfrak{l}}$ being compact and \mathfrak{l} semi-simple will be assumed in what follows – they hold in particular for the Lie algebra of traceless $N \times N$ anti-Hermitian matrices $\mathfrak{su}(N)$, which is of special interest as it corresponds to full controllability up to global phase.

Once we have introduced a desired parametrisation of the controls, the discretised propagator $U_{\mathbf{a}}(T)$ is a function mapping $\mathbb{R}^{RK} \rightarrow U(N)$, taking a vector \mathbf{a} composed of all parameters α_{rk} to the unitary matrix describing the evolution under the corresponding controls. Explicitly, $U_{\mathbf{a}}(T)$ equals the functional $U_{\mathbf{f}}(T)$ composed with the synthesis operator taking \mathbf{a} to the vector \mathbf{f} of functions such that $\mathbf{f}_r(t) = \sum_{k=1}^K \alpha_{rk} b_k(t)$. Solving $U_{\mathbf{a}}(T) = V$ can naturally be phrased as finding the root of either $U_{\mathbf{a}}(T) - V$ or $V^\dagger U_{\mathbf{a}}(T) - I$, but both of these functions range over a non-linear space, making them unsuitable for use in the Newton-Raphson algorithm.

Now let $\mathfrak{u}(N)$ denote the real linear space of all $N \times N$ anti-Hermitian matrices with inner product $\langle A, B \rangle = \text{Tr}(A^\dagger B)$. Amongst maps taking the unitary group to a linear space, the inverse $\log : U(N) \rightarrow \mathfrak{u}(N)$ of the well-studied exponential map is a complex analytic function whose value at I is 0 and derivative there is the identity on $\mathfrak{u}(N)$ – for $N > 2$, it is crucially the only such function taking the global phase neglecting subgroup $SU(N)$ to a linear space [11]. The most natural way to linearise our problem is therefore to look for a root of

$$\mathcal{L}(\mathbf{a}) = P \log(V^\dagger U_{\mathbf{a}}(T))$$

where the branches of the logarithm are chosen to give a result in $\mathfrak{l} + i\mathbb{R}I$ of minimal norm and P then expresses this in an orthonormal basis of \mathfrak{l} , dropping any iI component. Note that when \mathfrak{l} is $\mathfrak{su}(N)$, the choice of branches for each eigenvalue is the standard one with imaginary part in $(-\pi, \pi]$, while the implementation of P can just keep each strictly upper triangular entry of the input matrix and expresses the imaginary part of its diagonal in any orthonormal basis containing the vector of all $1/\sqrt{N}$. This choice for \mathcal{L} also has the advantage, when $e^{\mathfrak{l}}$ is compact, of making the corresponding error $\sqrt{\mathcal{E}(\mathbf{a})} = \|\mathcal{L}(\mathbf{a})\|$ equal the geodesic distance between $U_{\mathbf{a}}(T)$ and V over the group $e^{\mathfrak{l}}/U(1)$, ie. $e^{\mathfrak{l}}$ quotiented out by global phase (see [12] Sect. 3.2).

In its strong form [13], Sard's theorem gives that the set of target gates V in $e^{\mathfrak{l}}/U(1)$ for which a solution \mathbf{a}^*

to $U_{\mathbf{a}^*}(T) \equiv V$ exists with rank deficient Jacobian $d\mathfrak{L}|_{\mathbf{a}^*}$ is of Hausdorff dimension at most $m - 1$. Here m is still the dimension of the co-domain of \mathfrak{L} , namely \mathfrak{l} , which makes $m = N^2 - 1$ in the most prominent case when \mathfrak{l} is $\mathfrak{su}(N)$. Suppose the K , with $M = RK > m$, linearly independent basis functions b_k are not so degenerate that the image of \mathfrak{L} is a measure zero set. With respect to choosing a target V from the accessible set, of all $U_{\mathbf{a}}(T)$ for fixed T , the full rank condition implying quadratic convergence of the Newton-Raphson algorithm therefore occurs with probability one.

Computing the Jacobian

The chain rule gives a decomposition for each column

$$\frac{\partial}{\partial \alpha_{rk}} \mathfrak{L}(\mathbf{a}) = P d \log|_{V^\dagger U_{\mathbf{a}}(T)} \left(V^\dagger \frac{\partial}{\partial \alpha_{rk}} U_{\mathbf{a}}(T) \right) \quad (3)$$

of the Jacobian of \mathfrak{L} , with the derivative of the discretised propagator known to be

$$\frac{\partial}{\partial \alpha_{rk}} U_{\mathbf{a}}(T) = - \int_0^T U_{\mathbf{a}}(T) U_{\mathbf{a}}(t)^\dagger i H_r b_k(t) U_{\mathbf{a}}(t) dt \quad (4)$$

Writing $W = V^\dagger U_{\mathbf{a}}(T)$, given that the composition $\exp \circ \log$ is the identity on $U(N)$, the $d \log|_W$ part of (3) is simply $(d \exp|_{\log(W)})^{-1}$, which admits a closed form expression. Indeed, for a general anti-Hermitian matrix A , if λ_r are the eigenvalues of A and Λ a corresponding matrix of eigenvectors (one per column), $d \exp|_A(D)$ can be computed as $e^A \Lambda (\Gamma \cdot (\Lambda^\dagger D \Lambda)) \Lambda^\dagger$. Here the dot denotes elementwise multiplication and Γ is the matrix with entries $\Gamma_{rs} = \gamma(\lambda_s - \lambda_r)$ where γ is the function $z \mapsto \frac{e^z - 1}{z}$ continuously extended, so that $\gamma(0) = 1$. So $d \log|_W(B)$ is $\Lambda ((\Lambda^\dagger W^\dagger B \Lambda) / \Gamma) \Lambda^\dagger$ where division is carried out elementwise and the λ_r and Λ are those from the eigen-decomposition of $\log(W)$.

For computing the propagator $U_{\mathbf{a}}(T)$, we use a fixed time stepping scheme which for $s = 1, \dots, S$ successively evaluates the two point propagator $U_{\mathbf{a}}(t_s, t_{s-1})$, defined as $U_{\mathbf{a}}(t_s) U_{\mathbf{a}}(t_{s-1})^\dagger$, numerically where $t_0 = 0$, $t_S = T$, and $t_s - t_{s-1}$ is the same for each s . In the piecewise constant control parametrisation case, we let $S = K$ so that all the $U_{\mathbf{a}}(t_s, t_{s-1})$ are matrix exponentials – computing the eigen-decomposition of the constant Hamiltonian over each time interval (t_s, t_{s-1}) then renders the computation of both the propagators and their derivatives relatively inexpensive. In other cases, such as our Hermite function parametrisation, any general ODE solver could be used, but we found the Magnus-4 method [14] which is specialised for linear ODEs to be accurate for a smaller number of steps S than in particular the standard Runge-Kutta method, also of fourth order. To evaluate the integral from (4), given that we know $U_{\mathbf{a}}$ at the endpoints of each interval (t_{s-1}, t_s) a Lobatto quadrature rule is most

appropriate, particularly the fourth order rule in order to match the accuracy of the propagator $U_{\mathbf{a}}(t)$. This rule would approximate an integral $\int_{t_{s-1}}^{t_s} g(\tau) d\tau$ by the weighted sum

$$\frac{1}{t_s - t_{s-1}} \left[\frac{1}{6} g(t_{s-1}) + \frac{2}{3} g((t_{s-1} + t_s)/2) + \frac{1}{6} g(t_s) \right]$$

so that for the whole $\int_0^T g(\tau) d\tau$ each value $g(t_1), \dots, g(t_{S-1})$ enters twice. We also used polynomial interpolation, based on the value and first derivative at t_{s-1} and t_s , to evaluate $U_{\mathbf{a}}$ at the midpoint of each interval (t_{s-1}, t_s) , together with precomputed values for each b_k at the full set of quadrature nodes t_s and $(t_{s-1} + t_s)/2$.

Trust-region approach

As described earlier, as soon as \mathbf{a} is sufficiently close to a solution we should update \mathbf{a} to $\mathbf{a} + \mathbf{p}$, where \mathbf{p} is a solution to $J\mathbf{p} = -\mathfrak{L}(\mathbf{a})$, on all subsequent iterations for which the Jacobian $J = d\mathfrak{L}|_{\mathbf{a}}$ is full rank. However, we need an update rule which is effective in general, and for this purpose we have found in practise that far from a solution the trust region method readily delivers larger decreases in square error \mathfrak{E} than doing a line search could, so for this reason we focus on the former.

The trust region approach to the Newton-Raphson method consists in finding

$$\mathbf{p} = \underset{\mathbf{x}: \|\mathbf{x}\| \leq r}{\operatorname{argmin}} \|J\mathbf{x} + \mathfrak{L}(\mathbf{a})\|^2 \quad (5)$$

for some trust region radius r , with the situation when the unconstrained minimum can be achieved requiring that we find the solution to $J\mathbf{x} = -\mathfrak{L}(\mathbf{a})$ having smallest norm. In the over-determined case $m > RK$, independently of whether the original root finding problem is solvable, the equation $J\mathbf{x} = -\mathfrak{L}(\mathbf{a})$ generically admits no solution so that minimising the norm of the residual $J\mathbf{x} + \mathfrak{L}(\mathbf{a})$ is the best we can aim for. In this case one would solve the optimisation problem (5) directly as an instance

$$\underset{\mathbf{x}: \|\mathbf{x}\| \leq r}{\operatorname{argmin}} \mathbf{x}^T J^T J \mathbf{x} + 2\mathfrak{L}(\mathbf{a})^T J \mathbf{x}$$

of the so called trust region sub-problem, which consists in minimising a quadratic function $\mathbf{x}^T A \mathbf{x} + 2\mathbf{g}^T \mathbf{x}$ over a ball of radius r . Without loss of generality A can be taken symmetric and the ball centred at the origin, then for A semi-definite as in our situation a solution to the sub-problem can always be found of the form $\mathbf{x}_\lambda = -(A - \lambda I)^{-1} \mathbf{g}$ for some $\lambda \leq 0$. To find the appropriate λ , we use the higher order analogue of the method from [15], although for ease of implementation a general

convex optimisation package such as CVXOPT could be invoked to solve the sub-problem itself.

Otherwise when $m < RK$, we can restrict attention to \mathbf{x} orthogonal to the null space of J by expressing it as $\mathbf{x} = J^T \mathbf{y}$, reducing the problem to

$$\begin{aligned} \operatorname{argmin}_{\mathbf{y}: \|J^T \mathbf{y}\| \leq r} \|JJ^T \mathbf{y} + \mathcal{L}(\mathbf{a})\|^2 = \\ \operatorname{argmin}_{\mathbf{y}: \mathbf{y}^T M \mathbf{y} \leq r^2} \mathbf{y}^T M^2 \mathbf{y} + 2\mathcal{L}(\mathbf{a})^T M \mathbf{y} \end{aligned} \quad (6)$$

where the matrix $M = JJ^T$ is defined on the co-domain of \mathcal{L} . This last problem can be solved through the trust region sub-problem instance

$$\operatorname{argmin}_{\mathbf{z}: \|\mathbf{z}\| \leq r} \mathbf{z}^T M \mathbf{z} + 2\mathcal{L}(\mathbf{a})^T \sqrt{M} \mathbf{z}$$

then using any solution of $\sqrt{M} \mathbf{y} = \mathbf{z}$, all of them being equivalent in that they yield the same final \mathbf{x} . Such a reformulation is advantageous since it makes the corresponding matrix A lower dimensional, and importantly for performance, instances of \sqrt{M} in the algorithm we use appear in such a way that it never needs to be computed.

In adapting the choice of trust region radius r , we strive on each iteration to use the value r_0 of r for which $\mathfrak{E}(\mathbf{a} + \mathbf{p}_r)$ attains its minimum, where \mathbf{p}_r is the solution from (5) with radius r . Note that by definition any choice of radius r above the norm r^* of the least square solution to $J\mathbf{x} = -\mathcal{L}(\mathbf{a})$ is equivalent, moreover along $\mathbf{a} + \mathbf{p}_r$ the model for \mathfrak{E} , namely $\|\mathcal{L}(\mathbf{a}) + J\mathbf{p}_r\|^2$, is strictly decreasing as r ranges over $[0, r^*]$. As long as the model for $\mathcal{L}(\mathbf{a} + \mathbf{p}_r)$ is accurate, the true value of \mathfrak{E} along $\mathbf{a} + \mathbf{p}_r$ must track its model value and therefore be decreasing – but once r is large enough that the model for \mathcal{L} starts to break down in the vicinity of $\mathbf{a} + \mathbf{p}_r$, the increments $\mathbf{p}_{r+\delta} - \mathbf{p}_r$ quickly become meaningless, making them overwhelmingly likely to lead the true $\mathfrak{E}(\mathbf{a} + \mathbf{p}_r)$ to increase. This causes the relative error of the model for $\mathfrak{E}(\mathbf{a} + \mathbf{p}_r) - \mathfrak{E}(\mathbf{a})$ to undergo a swift transition from small, in fact vanishing as $r \rightarrow 0$, to large magnitudes as r grows past the minimiser r_0 . In our implementation, we adjusted r to make this relative error satisfy

$$0.2 \leq -\frac{\|\mathcal{L}(\mathbf{a}) + J\mathbf{p}_r\|^2 - \mathfrak{E}(\mathbf{a})}{|\mathfrak{E}(\mathbf{a} + \mathbf{p}_r) - \mathfrak{E}(\mathbf{a})|} \leq 0.3$$

which typically places r close to r_0 , although the choice is a valid one irrespectively (see [1] Sect. 4.0).

Norm dependent structure

At the null control vector ie. $\mathbf{a} = \mathbf{0}$, the propagator $U_{\mathbf{a}}(t)$ reduces to the matrix exponential $e^{-iH_0 t}$, so that working in an eigenbasis of H_0 it is easy to see that

within any $U_{\mathbf{a}}(t)^\dagger H_r U_{\mathbf{a}}(t)$ expression from (4), each diagonal entry will be a constant function. Therefore over $\mathfrak{su}(N)$, only R out of $N - 1$ possible linear combinations of diagonal entries can be generated by any integral $\int_0^T U_{\mathbf{a}}(t)^\dagger H_r b(t) U_{\mathbf{a}}(t) dt$ where b is a scalar valued function. Hence the Jacobian at null controls of $U_{\mathbf{a}}(t)$, thus also of \mathcal{L} , will be rank deficient for non-trivial systems, since it would be unrealistic for a system to have $R \geq N - 1$ controls unless it were of very low dimension N . Then the model for \mathcal{L} at $\mathbf{a} = \mathbf{0}$ almost surely admits no exact solution, so that the ill-conditioning defined as $\|J^T(JJ^T)^{-1}\mathcal{L}(\mathbf{a})\|$ is effectively infinite there, and by continuity tends to infinity as $\|\mathbf{a}\| \rightarrow 0$.

At the other extreme, \mathbf{f} being large introduces high frequency oscillation in $U_{\mathbf{f}}(t)$ and $U_{\mathbf{f}}(t)^\dagger H_r U_{\mathbf{f}}(t)$, which cancels out when integrating $U_{\mathbf{f}}(t)^\dagger H_r b_k(t) U_{\mathbf{f}}(t)$ for any fixed basis elements b_k . In other words, eigenfunctions of the infinite dimensional Jacobian $dU_{\mathbf{f}}(T)$ have a lot of their spectrum in high Fourier components, and this is lost when restricting to the lower frequency subspace spanned by the b_k to obtain the finite dimensional J . As a consequence the singular values of the discretised Jacobian J shrink as $\|\mathbf{a}\|$ grows, with the corresponding ill-conditioning almost surely going to infinity. This explains why the ill-conditioning curve from Fig. 2(a) grows towards both small and large norms, thereby attaining its minimum at some finite norm value.

For any given target gate V , there is a minimal norm μ below which no solution in the chosen basis can be found to the control problem, up to the tolerated error. Due to the high dimensionality RK of the discretised control space, the volume of parameter vectors \mathbf{a} below some norm x increases extremely quickly with x , eg. for our test problem increasing x by 10% will make the volume grow by a factor of over 10^{82} . While it is tautological that any successful algorithm run must terminate with $\|\mathbf{a}\|$ above μ , there should be no shortage of solutions with norms slightly above μ , so we can expect the final norm to be close to μ when starting with $\|\mathbf{a}^{(0)}\| < \mu$. When the norm of any iterate $\mathbf{a}^{(n)}$ is above μ , it is reasonable to assume the update $\mathbf{a}^{(n+1)} - \mathbf{a}^{(n)}$ has no preferred direction, which by the high dimensionality would imply it is near orthogonal to $\mathbf{a}^{(n)}$ with high probability. Since the trust region radius r , hence the update, should be noticeably smaller than the current iterate $\mathbf{a}^{(n)}$ by a factor $1/\rho \gg 1$ say, we would conclude that $\|\mathbf{a}^{(n+1)}\|$ is only a factor of $1 + \rho^2/2$ greater than $\|\mathbf{a}^{(n)}\|$. Therefore when starting with $\|\mathbf{a}^{(0)}\| > \mu$, none of the algorithm iterations change the norm of \mathbf{a} substantially, making the final norm close to the initial norm. These considerations account for the characteristic shape of the curve in Fig. 2(c), which matches the identity function down to some floor level, presumably equal to μ , below which it hovers just above the floor level.

Given that the ill-conditioning measures how difficult

it is to decrease the error on a given iteration, when $\|\mathbf{a}^{(0)}\| > \mu$ and all iterates \mathbf{a} have the same norm, runs of the algorithm are faster if and only if the ill-conditioning is lower for this norm. Otherwise the norm of iterates \mathbf{a} increases up to some value above μ , but since the ill-conditioning curve is increasing below μ , lower initial norms lead to runs being slower. The runtime does not however blow up as $\|\mathbf{a}^{(0)}\| \rightarrow 0$ because even starting from $\mathbf{0}$, although the first iteration may only reduce the error by a negligible amount, the resulting $\|\mathbf{a}^{(1)}\|$ equal to the trust region radius will be substantial. Finally, this explains the correspondence in Fig. 2 between the runtime curve and ill-conditioning curve.

Test problem

The system used for numerical illustrations in this letter was a chain of five qubits with nearest neighbour Ising coupling, with a linear gradient inhomogeneity in the magnetic field to enable some degree of frequency selective addressing. Explicitly, the intrinsic Hamiltonian is

$$H_0 = \sum_{n=1}^4 \sigma_z^{(n)} \sigma_z^{(n+1)} - \sum_{n=1}^5 \omega_n \sigma_z^{(n)}$$

with frequencies $\omega_n = n+2$, and where $\sigma_z^{(n)}$ is the Pauli z matrix acting on the n^{th} spin, eg. $\sigma_z^{(2)} = I \otimes \sigma_z \otimes I \otimes I$. The control Hamiltonians are

$$H_1 = \sum_{n=1}^5 \sigma_x^{(n)}, \quad H_2 = \sum_{n=1}^5 \sigma_y^{(n)}$$

corresponding to the x -coordinate and y -coordinate components \mathbf{f}_1 and \mathbf{f}_2 of an electric pulse applied simultaneously to all qubits. For this problem, the total evolution time is fixed at $T = 125$, and the number of basis functions (per control) is chosen to be $K = 1000$, with all data in the first three figures coming from using piecewise constant controls. By piecewise constant, we formally mean that $b_1(t)$ is vanishing for t outside the interval $(0, T/K)$ and constant equal to one inside, with each b_k equal to the previous b_{k-1} translated forward in time by T/K . The Hermite functions refer to the eigenfunctions of the quantum harmonic oscillator, shifted to be centred at $T/2$, and jointly scaled about $T/2$ so that the maximum any of the first K attain outside $(0, T)$ is 10^{-8} . Moreover, the definition of integrated power norm for the control vector \mathbf{f} we use satisfies

$$\|\mathbf{f}\|^2 = \sum_{r=1}^R \int_0^T |\mathbf{f}_r(t)|^2 dt$$

which equals the standard Euclidian norm of the parameter vector \mathbf{a} when the basis functions b_1, \dots, b_K are orthonormal.

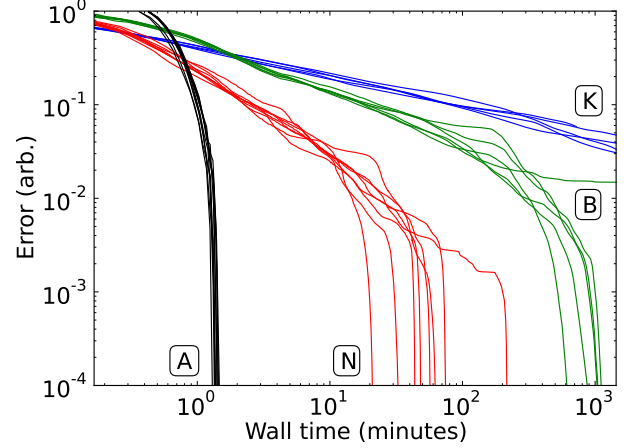


FIG. 5: Corresponding to Fig. 1 from the main text, here for our second test problem. Performance comparison between several runs of the Newton-Raphson (red, labeled ‘N’), BFGS GRAPE (green, labeled ‘B’) and Krotov (blue, labeled ‘K’) algorithms with moderately sized initial pulses $\|\mathbf{f}^{(0)}\| = 20$. Also shown (in black, labeled ‘A’) are Newton-Raphson runs preceded by a routine to find the norm with least ill-conditioning.

As a second test problem, we can consider implementing a logical T-gate encoded with the five physical qubit stabilizer code as described in [16]. The underlying system is a Heisenberg spin chain of length five, with a fixed external coupling field at a Rabi frequency of 10, so that H_0 reads

$$\sum_{n=1}^4 \sigma_x^{(n)} \sigma_x^{(n+1)} + \sigma_y^{(n)} \sigma_y^{(n+1)} + \sigma_z^{(n)} \sigma_z^{(n+1)} + 10 \sum_{n=1}^5 \sigma_x^{(n)}$$

with an evolution time $T = 90$. This has a single control corresponding to $H_1 = \sigma_z^{(1)}$ enabling the first spin to be detuned, through a local voltage which is piecewise constant over $K = 1500$ intervals.

-
- [1] J. Nocedal and S. J. Wright. *Numerical Optimization*. (Springer, New York, 1999)
 - [2] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquieres, A. Gruslys, S. G. Schirmer, and T. Schulte-Herbrüggen. *Phys. Rev. A*, **84**, 022305 (2011)
 - [3] A. Griewank. *Math. Prog.*, **50**, 141 (1991)
 - [4] M. J. D. Powell. *P. Int. Congr. Math.*, 1525 (1983)
 - [5] M. J. D. Powell. *Math. Prog.*, **11**, 42 (1976)
 - [6] D. C. Liu and J. Nocedal. *Math. Prog.*, **45**, 503 (1989)
 - [7] H. Akaike. *Ann. Inst. Stat. Math.*, **11**, 1 (1959)
 - [8] A. Ben-Israel. *J. Math. Anal. & Appl.*, **15**, 243 (1966)
 - [9] S. Helgason. *Differential geometry, Lie groups, and symmetric spaces*. (Academic Press, New York, 1981)
 - [10] V. Jurdjevic and H. J. Sussmann. *J. Diff. Eq.*, **12**, 313 (1972)

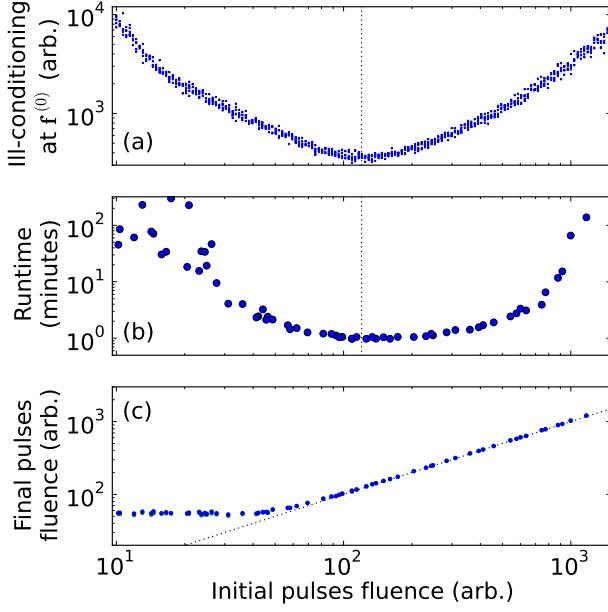


FIG. 6: Direct analogue for our second test problem of Fig. 2 from the main text. For Newton-Raphson runs with initial pulses $\mathbf{f}^{(0)}$ of different norms, (b) the wall time needed to reach an error ε of 10^{-4} and (c) the norm of the corresponding solution pulses, with a dashed ‘initial equals final’ line. In addition, (a) the ill-conditioning of the Jacobian at several randomly sampled pulses of each norm.

- [11] F. Kang and S. Zai-jiu. *Numerische Mathematik*, **71**, 451 (1995)
- [12] A. Arvanitogeórgos. *An introduction to Lie groups and the geometry of homogeneous spaces*. (AMS Bookstore, 2003)
- [13] H. Federer. *Geometric measure theory*. (Springer, 1996)
- [14] A. Iserles, S. P. Nørsett, and A. F. Rasmussen. *Appl. Numer. Math.* **39**, 379 (2001)
- [15] J. J. Moré and D. C. Sorensen. *SIAM J. Sci. & Stat. Comp.*, **4**, 553 (1983)
- [16] R. Nigmatullin and S. G. Schirmer. *New J. Phys.*, **11**, 105032 (2009)
- [17] <http://scikits.appspot.com>